

REMARKS

This Amendment and Response is filed in connection with the Office Action mailed on March 13, 2008. Please consider the above-identified patent application in view of the following.

No claims are amended herein, no claims are canceled, and no claims are newly added; as a result, claims 1, 3-7, 15, 17-22, and 24-28 are pending in this application.

Examiner Interview Summary

The Applicant would like to thank Examiner Ben C. Wang for the courtesy of a phone interview on May 20, 2008 between the Examiner, the Applicant's representative, Jim H. Salter, and inventor, Robert M. Zeidman. During the interview, the claims and the cited references were discussed and an agreement was reached that the Examiner would consider the references in view of the arguments provided below.

§103 Rejection of the Claims

Claims 1, 3, 5, 7, 15, 17, 19, 21-22, 24, 26, and 28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Lehman, et al. (U.S. 4,796,179, hereinafter, "Lehman") in view of Gauthier et al., "Automatic Generation and Targeting of Application Specific Operating Systems and Embedded Systems Software", 2001, IEEE (hereinafter 'Gauthier'), and further in view of Stewart et al., 'The Chimera Methodology: Designing Dynamically Reconfigurable Real-Time Software Using Port-Based Objects', 1995, IEEE (hereinafter, 'Stewart').

Claims 4, 18, and 25 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Lehman, in view of Gauthier and further in view of Xu et al., "On Satisfying Timing Constraints in Hard-Real-Time Systems", 1991, ACM (hereinafter 'Xu').

Claims 6, 20, and 27 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Lehman, in view of Gauthier and Xu, and further in view of D. Lake (US 2004/0045003A1) (hereinafter 'Lake').

Applicant respectfully submits that the Office Action did not make out a *prima facie* case of obviousness, because even if combined, the cited references fail to teach or suggest all of the claim limitations of the independent claims of the present Application.

Applicant continues to assert that the previously submitted claims with the recited limitations distinguish over Lehman and all other art of record. As correctly admitted in the Office Action at page 3, “Lehman does not explicitly disclose … synthesizing source code from commands embedded in source code to implement a task scheduler that uses the scheduling algorithm for controlling execution of said n tasks, said synthesized source code being executable on a target system after compilation.” Lehman does not use embedded commands. Rather, Lehman provides functional blocks where each functional block has a corresponding source code template which is used to generate the source code. Each template in Lehman includes invariant code which defines a computation, and variables for tailoring the source code to use any specified parameters and also to couple the source code to the memory locations for its inputs and outputs. The present invention as currently claimed does not use or need templates or invariant code. As such, Lehman fails to teach each and every element of the currently pending claims.

Gauthier describes a method for the automatic generation of application specific operating systems and automatic targeting of application software. At section 3.5.3 of Gauthier, the reference states, “Code Expander takes as input a list of macro code from Code Selector and parameters (processor and allocation information[]) from Architecture Analyzer. It generates the final OS code by expanding the macro codes of elements to source codes (in C or assembly).” However, Gauthier also does not describe or suggest an implementation as presently claimed for specifying two types of tasks, one type that can be scheduled for execution using a specified scheduling algorithm, and another type of task that is executed only once upon the initial execution of the task scheduler. As correctly admitted in the Office Action at page 5, “Lehman and Gauthier do not explicitly disclose, “specifying t init-tasks that are executed only once upon initial execution of a task scheduler, t being less than or equal to n... the task scheduler further controlling one execution of each of said set of t init-tasks.”

Stewart describes a software assembly system for rapid development of real-time applications through use of dynamically reconfigurable and reusable software. In the Office Action on the bottom of page 5, it is stated that Stewart discloses the claimed init tasks. However, as explained during the Examiner interview, Stewart only describes each task

as having a portion of the task dedicated for initialization code that is executed only the first time the task is called. As such, the initialization code in Stewart is not an independently schedulable task. Stewart does not disclose or suggest a schedulable initialization task (i.e. the claimed init-task) that gets executed only once at initialization. Further, because the initialization code in Stewart is only a portion of each task, there is a one-for-one correspondence in Stewart between the number of portions of initialization code and the number of tasks. As such, in Stewart there are n portions of initialization code if there are n tasks.

In the present invention as currently claimed, for example in claim 1, the claimed invention includes, “specifying t init-tasks that are executed only once upon initial execution of a task scheduler, t being less than or equal to n ... the task scheduler further controlling one execution of each of said set of t init-tasks.” Thus as claimed, the init-task in the present invention is an independently schedulable task executed once by the task scheduler. It is clear from the currently pending claims and from the specification filed with the present application that the claimed init-task is a task of a type that can be scheduled by the task scheduler and is not merely a portion of code. In the specification filed with the present application, page 1, lines 13-20, each task is allocated a time slice that is allocated according to a scheduling algorithm and a priority and controlled by a kernel of a real-time operating system. Thus, the task as claimed in the pending claims and described in the specification filed with the present application is an independently schedulable entity under control of a task scheduler that can be distinguished from the initialization code portion in Stewart that is only a portion of each task and not itself an independently schedulable entity.

Further, as currently claimed, the number of init tasks t is less than or equal to the number of tasks n . In the presently claimed invention, there can be fewer init-tasks because the currently claimed init-tasks are independently schedulable entities under control of a task scheduler. In contrast, because the initialization code in Stewart is only a portion of each task, there is a one-for-one correspondence in Stewart between the number of portions of initialization code and the number of tasks. As such, in Stewart there are n portions of initialization code if there are n tasks. For example, see Stewart at

page 50, bottom left paragraph, where Stewart states, “Every port-based object... has the structure shown in Figure 7.” Figure 7, shown in Stewart and referenced in the Office Action, shows an init block or portion of code; but the cited part of Stewart does not show an independently schedulable init-task as currently claimed.

As explained above, Stewart fails to describe or suggest elements of the currently pending claims. Additionally, Stewart teaches away from the present invention. In Stewart on page 46, section 1, the reference states (emphasis added),

“This process of assembling an application **without writing or automatically generating any new glue code** is called *software assembly*. In this paper, we describe the Chimera methodology, which defines new software models, communication protocols, and interfaces for supporting software assembly.”

In the last paragraph on page 46, Stewart states,

“Software synthesis, also known as automatic code generation... generate[s] the ‘glue’ code for automatically integrating reusable modules. As input they receive information about the software modules, the interface specifications and the target application, and as output produce code using both formal computations and heuristics. For a truly generic framework, however, it is desirable that the integration of software be based on the interfaces alone, and not on the semantics of the modules or application, as the latter results in the need for large knowledge bases. Furthermore, software synthesis only allows for statically configuring an application, and usually does not support dynamic reconfiguration.”

Therefore, Stewart offers an alternative to software synthesis called software assembly that is supposedly better and distinguished from the software synthesis as described and claimed in the present application. As evident in the title of the present invention (i.e. “SOFTWARE TOOL FOR SYNTHESIZING A REAL-TIME OPERATING SYSTEM”) and as stated, for example at page 5, lines 134-135 of the specification of the present application, “[t]he present invention provides a “synthesized” RTOS that automates the process of creating an RTOS.” As such, Stewart expressly teaches away from the present invention and thus cannot be used to render the presently claimed invention obvious.

Therefore, the Applicant respectfully submits that at least for the reasons set forth above, independent claims 1, 15, and 22 and their dependent claims are allowable over Lehman, Gauthier, Stewart, Xu, Lake, the other cited references, and combinations therof. The Applicant respectfully submits that the current rejections have been overcome. The Applicant respectfully requests withdrawal of the §103(a) rejections.

CONCLUSION

Applicant respectfully submits that the claims are in condition for allowance, and notification to that effect is earnestly requested. The Examiner is invited to telephone Applicant's attorney, Jim H. Salter at 408-406-4855 to facilitate prosecution of this application.

Respectfully submitted,

ROBERT M. ZEIDMAN

By his Representatives,

Salter Intellectual Property Law
105 Thoreau Lane
Folsom, CA 95630
408-406-4855

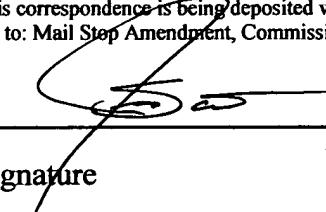
Date May 27, 2008

By 
Jim H. Salter
Reg. No. 35,668

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: Mail Stop Amendment, Commissioner of Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on this 27 day of May, 2008.

Jim H. Salter

Name


Signature